

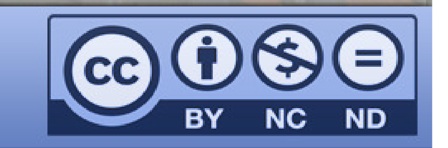
Bien débuter son développement pour ArcGIS,

Partie 2, Professionnaliser son code

N. Py

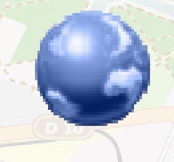
Journées ESRI
SIG2010, Octobre 2010





Bien débuter son développement pour ArcGIS,

Partie 2, Professionnaliser son code



Contexte

Phase de développement: des principes

Phase de développement: des pratiques

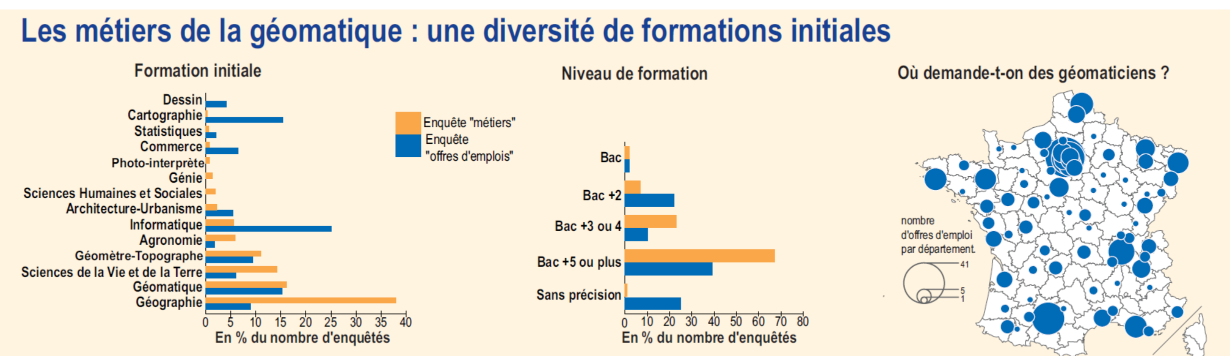
Phase de développement: des outils généraux (et gratuits)

Conclusion



Le Géomaticien, de + en + informaticien

Les métiers de la géomatique : une diversité de formations initiales



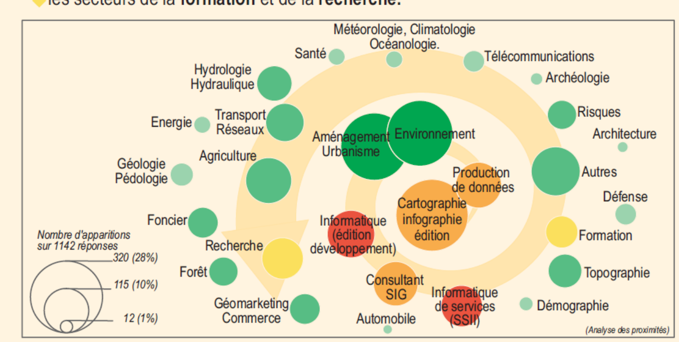
Un décalage entre l'activité observée et les spécialisations recherchées

- Plus de 35% des **emplois sont occupés** par des **"géographes"**, un quart des **offres d'emplois** demande des **"spécialistes en informatique"** et une offre sur six concerne un **"cartographe"**.
- L'identité de **"géomaticien"** apparaît fortement à parts égales dans les 2 enquêtes (15%).
- On observe deux principaux niveaux de recrutements : **techniciens et ingénieurs**, avec un avantage pour les formations de haut niveau (bac+5) et une inversion de rapport entre ces 2 niveaux de formation pour l'offre d'emplois et les métiers occupés.

Des emplois à l'intersection de plusieurs secteurs d'activités : du thématique à la technique

Les secteurs d'activités de la géomatique s'organisent principalement autour de :
- **la production, la représentation de l'information géographique**,
- les **développements informatiques**,
- les thématiques de **l'environnement et de l'aménagement.**

En périphérie de ce noyau central:
- plusieurs **autres thématiques:** les **risques**, **l'agriculture**, la **forêt**, ...
- des **nouveaux domaines** d'application: la **santé**, les **télécommunications**, ...
- les secteurs de la **formation** et de la **recherche.**



Un métier centré sur le développement de SIG et la cartographie

◇ Dans 45% des cas, l'activité consiste à **développer et mettre en place un SIG** et la **cartographie** est une des composantes dans 40% des cas.

◇ Deux groupes d'activités apparaissent dans 20% à 30% des emplois occupés :
- **conception et gestion** de bases de données liées aux SIG,
- **acquisition** à l'amont et **traitement** de l'information à l'aval.

◇ Enfin, plus spécifiques, la **gestion de projets**, les **développements informatiques** et l'**analyse spatiale**.

Les activités du géomaticien	En % du nombre d'enquêtes
Mise en place - développement et administration d'un SIG	46.1
Cartographie (CAO - DAO - infographie - édition ...)	38.9
Acquisition/Intégration de données (non topographiques)	30.4
Gestion et administration de base de données	28.2
Conception et développement de BD	26.5
Numérisation/Digitalisation/Saisie	22.7
Echanges de données/Relation avec partenaires ou prestataires	22.3
Traitements et analyses statistiques	20.4
Formation et assistance aux utilisateurs	19.2
Conduite ou gestion de projets - suivi budgétaire	18.7
Modélisation et analyses spatiales	18.3
Cartographie Web - développement et applications Web	15.8
Programmation - développement d'applicatifs métiers	15.3

Source: http://georezo.net/wiki/_media/main:formetiers:poster_metier_mathian_2003_2.pdf

Avec cependant des lacunes en Génie logiciel…

1. **Baisse d'emploi des langages de script propriétaire:**
   1. **Avenue**, (AV 3.x, 1996 ; v3.3 en 'Mature support' depuis 2002)
   2. **AML** (AI v4, 1986 ; AI 9.3.1 = AI 9.3 en 'General availability depuis 2008)
   3. **Fin du support VBA après la v9.4/10** (*VBA will be available if needed, fully supported but not recommended.*)
   4. **'Remplacement' par desktop add-ins en v9.4/10**
2. **ArcObjects** (AG 8.x, 1999 ; AG 9.3.1 en 'General availability depuis 2009)
3. **Model Builder** (AG 9.x, 2004 ; AG 9.3.1 en 'General availability depuis 2009)
4. **Python** (AG 9.x, 2004 ; AG 9.3.1 en 'General availability depuis 2009)
5. **Support Web**
   1. ArcIMS (AIms 4, 2002 ; 9.3.1 en 'General availability depuis 2009)
   2. Web ADF (Java/.Net) (9.2, 2006)
   3. APIs Riches (9.3, 2008)
6. **Suppport smartphone**
   1. IPhone
   2. Blackberry
   3. Androïd

Diversification et 'complexification' des outils



ESRI et le développement à façon, en image…

Platform Performance Change

>> 10x Faster Hardware

>> 10x Faster Hardware

> 2x Process Load
>> 10x Faster Technology Change

> 2x Process Load
>> 10x Faster Technology Change

Tightly Scripted Software Code
1982–1999

Object Relational Software

Service Oriented Architecture
Hardware Virtualization
Cloud Computing

ARC/INFO 1982
ArcView GIS 1992
MapObjects 1996
ArcIMS 1997

ArcGIS Desktop
- ArcInfo 1999
  - ArcEditor 2000
  - ArcView
- ArcGIS Server 2004
- ArcGIS Engine

- ArcGIS Server 2008 -2009
  - REST API
  - WEB 2.0 Interoperability
  - User Publishing / Collaboration
  - ArcGIS Online Resources'
  - Rich Internet Clients
    - Javascript, Flex API, Silverlight

Coverages
Librarian
Shapefiles
ArcStorm
Spatial Database Engine (SDE)
- Simple non-versioned file types

ArcSDE 1999
- Geodatabase Versioning (ArcSDE)
  - Distributed Geodatabase Replication

ArcGIS Server 2009
-Faster Maps
-Map Cache
-Map Service Description (MSD)
-Optimized Mapping tools

Scripts → Objects → Services

Source: http://www.wiki.gis.com/wiki/index.php/File:SDS10Fig2.1_SWEvolution.jpg

Plus généralement en géomatique

1. Open source se démocratise
2. Et augmente en fonctionnalités
3. Augmentation des besoins, rejoignant la vision GeoDesign ESRI
    1. Plus seulement de la représentation
    2. Mais de la modélisation
    3. Et de la prédiction
4. Augmentation des volumes de données traitées
5. Multiplication des logiciels employés
6. Demande d'automatisation des tâches en hausse

Objectif de la présentation

Professionnaliser son code .Net

1. L'acquisition de concepts et vocabulaire
2. Un aperçu théorique des méthodes des développeurs
3. La visualisation de quelques outils, dans un contexte .Net

Grâce à mon retour d'expérience
• De non-développeur de formation…
• Acquise sur le tas
• Au sein d'un établissement plutôt open source et agile

Bien débuter son développement pour ArcGIS,

Partie 2, Professionnaliser son code

Contexte



Phase de développement: des principes

Phase de développement: des pratiques

Phase de développement: des outils généraux (et gratuits)

Conclusion

Principes de développement, en vrac …

- KISS : Keep It Small & Simple

(http://en.wikipedia.org/wiki/KISS_principle)

- DRY : Don't Repeat Yourself

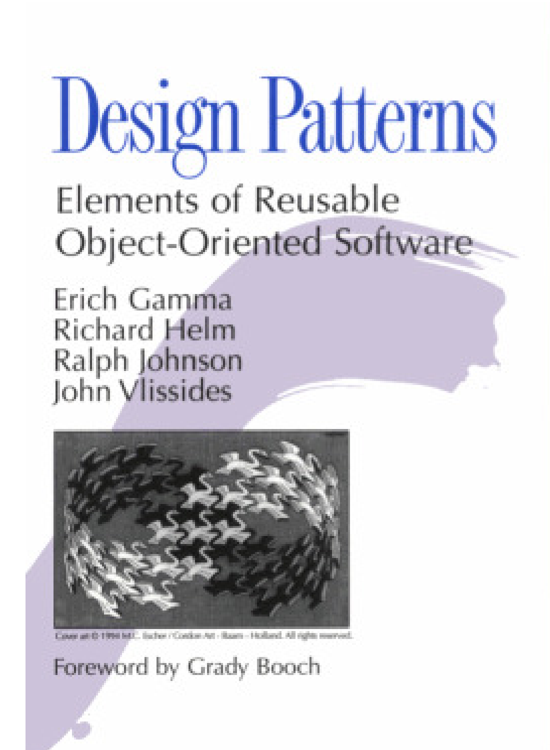
(http://en.wikipedia.org/wiki/Don%27t_repeat_yourself)

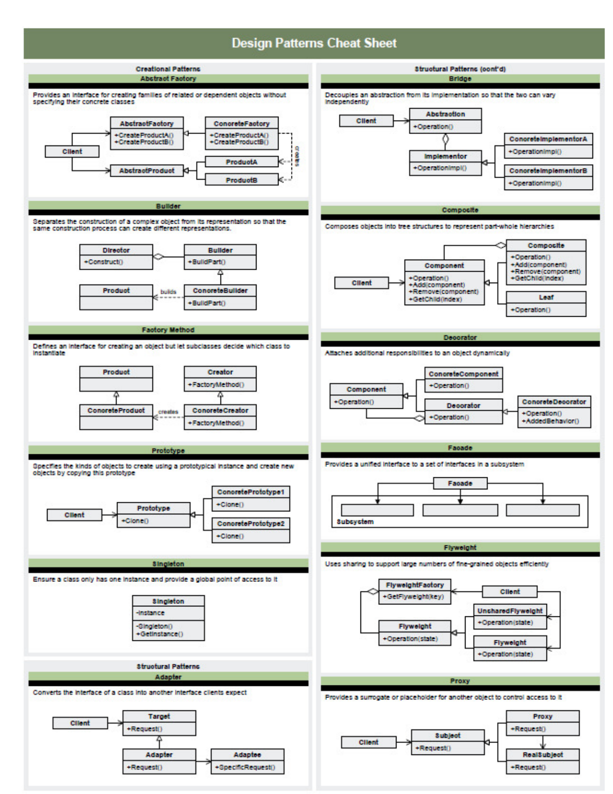
- YAGNI : You Ain't Gonna Need It

(n'ajouter une fonctionnalité que si nécessaire) (http://en.wikipedia.org/wiki/You_Ain%27t_Gonna_Need_It)

Les design patterns (patrons de conception)

- décrivent des solutions standard
- pour répondre à des problèmes d'architecture
- dans le monde objet





Source: http://blog.acodingfool.com/wp-content/uploads/2009/07/design_pattern_cheatsheet.pdf



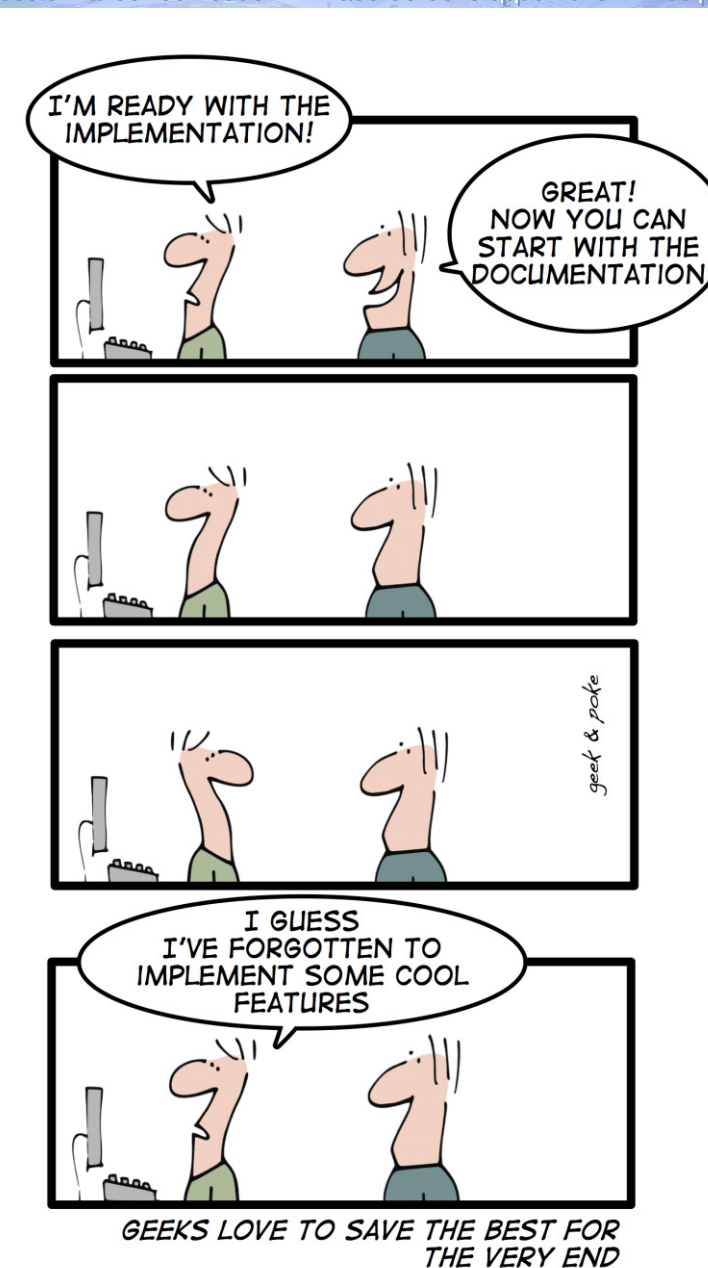
Principes de développement, nommage

- Standardiser la façon de nommer:
  - Les variables
  - Les classes
  - …
- Pourquoi ?
  - Code source + lisible,
  - Apporte des métadonnées(objectif, objet concerné,…)
  - Maintenance automatisable
  - …



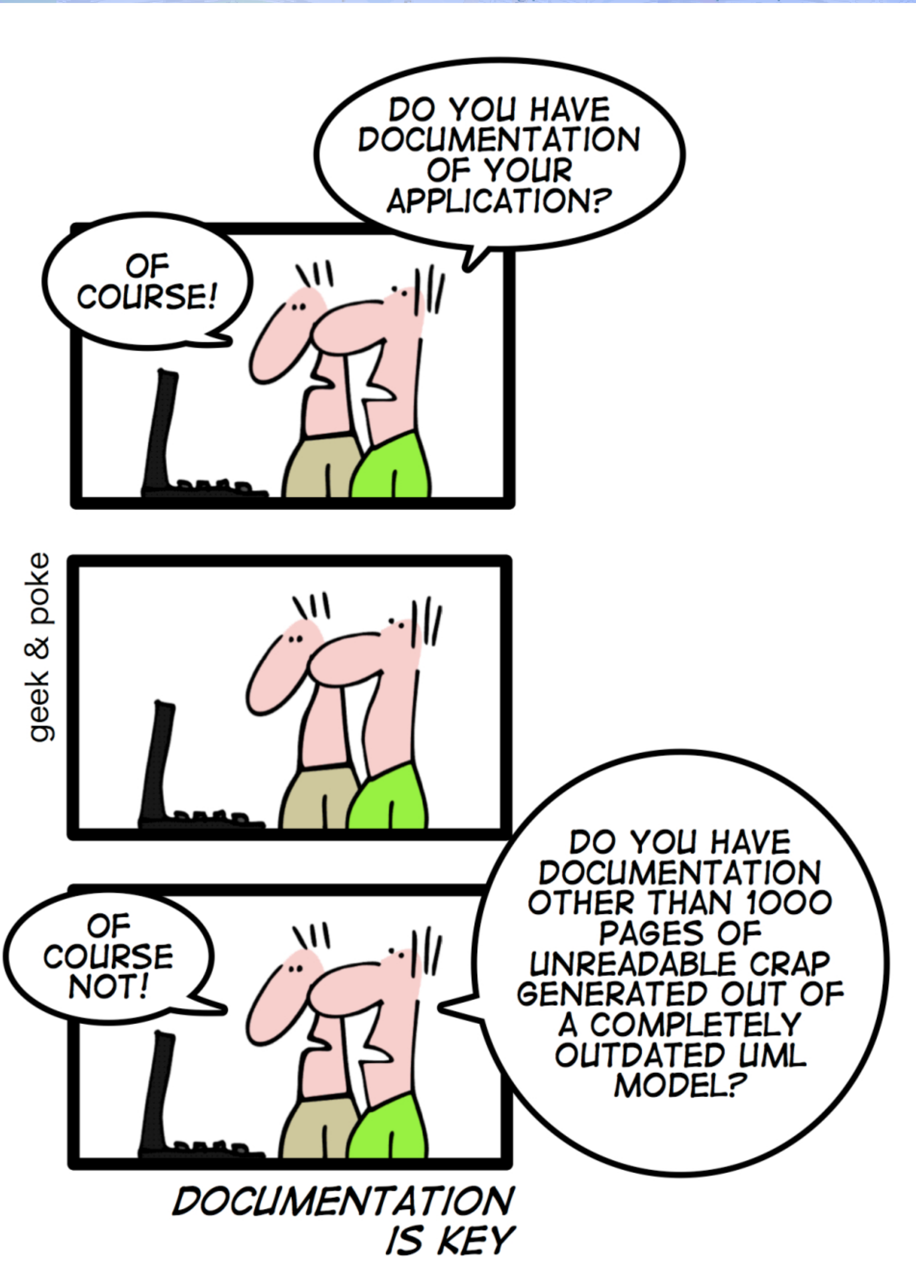
Principes de développement, documentation du projet

Source: http://geekandpoke.typepad.com/geekandpoke/2009/06/what-geeks-love-part-2.html



- Plusieurs types de documentation:
- Besoins(cahier des charges)
  - Fonctionnalités
  - Performances
  - Sécurité
  - …
- Communication avec le logiciel (inputs, outputs)
- Architecture (logicielle, physique,…)
- Comptes rendus de réunions
- Cahiers de tests (vierges et passés)
- Marketing
- **+ doc du logiciel**

Principes de développement, documentation du logiciel



Source: http://geekandpoke.typepad.com/geekandpoke/2010/01/documentation-is-key.html

- Technique:
  - Choix (// contraintes)
  - pb rencontrés,
  - Solutions,
  - …
- Administration
  - manuel d'installation,
  - de maintenance
- Manuel utilisateur
- Dictionnaire des classes
- Librairies tierces (+ code source)

Bien débuter son développement pour ArcGIS,

Partie 2, Professionnaliser son code

Contexte

Phase de développement: des principes

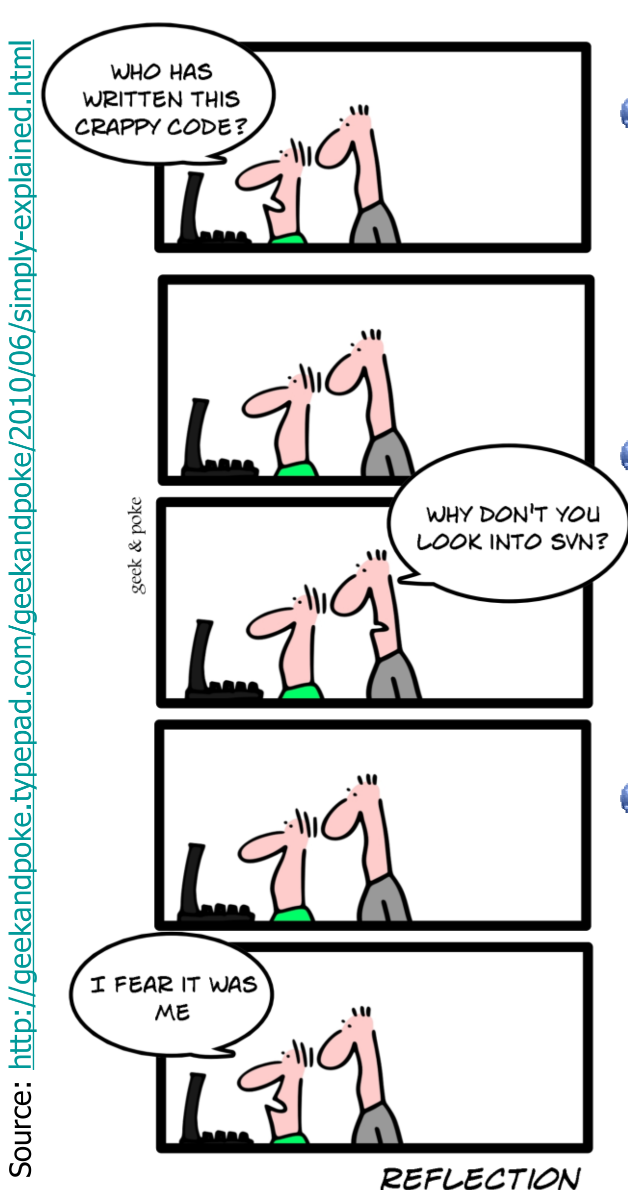


Phase de développement: des pratiques

Phase de développement: des outils généraux (et gratuits)

Conclusion

La gestion de versions, quoi ?



Source: http://geekandpoke.typepad.com/geekandpoke/2010/06/simply-explained.html

- Dans le développement, on modifie le code:
  - Ajouts,
  - Modifications,
  - Suppressions.
- La gestion de version enregistre ces modifications
  - Datées,
  - Commentées,
  - Associées à un développeur.
- Elle permet donc notamment:
  - La restauration de l'état d'un projet à une date/modification donnée,
  - D'assurer une sauvegarde du projet.

La gestion de versions, quand ?

- Toujours
- Versionnement + backup
- Coût <<< bénéfices

- Chez ESRI:
  - ESRI DS2010 Developer-to-Developer: The ESRI Development Process (http://proceedings.esri.com/library/userconf/devsummit10/papers/tech/source_code_and_build.pdf)

La gestion de versions, l'exemple de SVN+Tortoise (1/2).

SUBVERSION

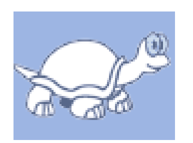
- Subversion (SVN)

Exposition web

WebSVN

GUI

RapidSVN



TortoiseSVN

- Client open source gratuit pour SVN
- Intégration dans l'explorateur windows
- Recouvrement d'icônes
- Accès facile aux commande de Subversion (évite les lignes de commande)



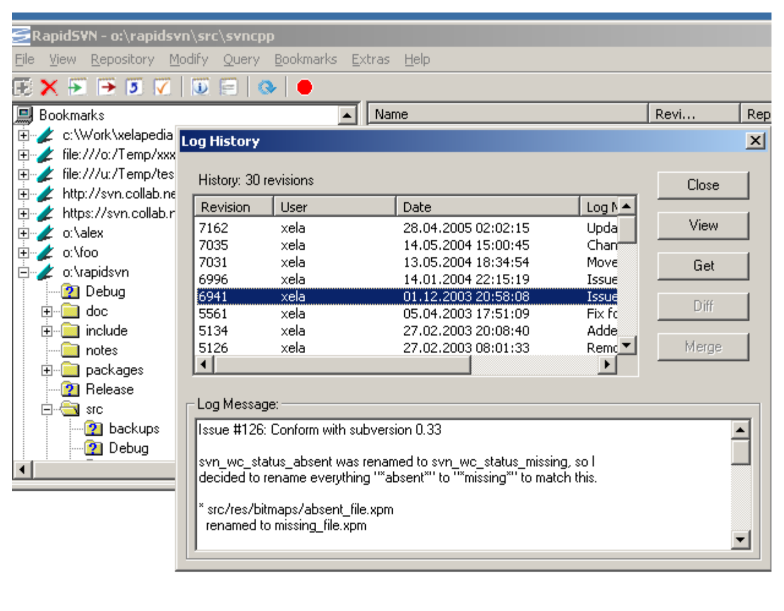
La gestion de versions, l'exemple de SVN+Tortoise (2/2).

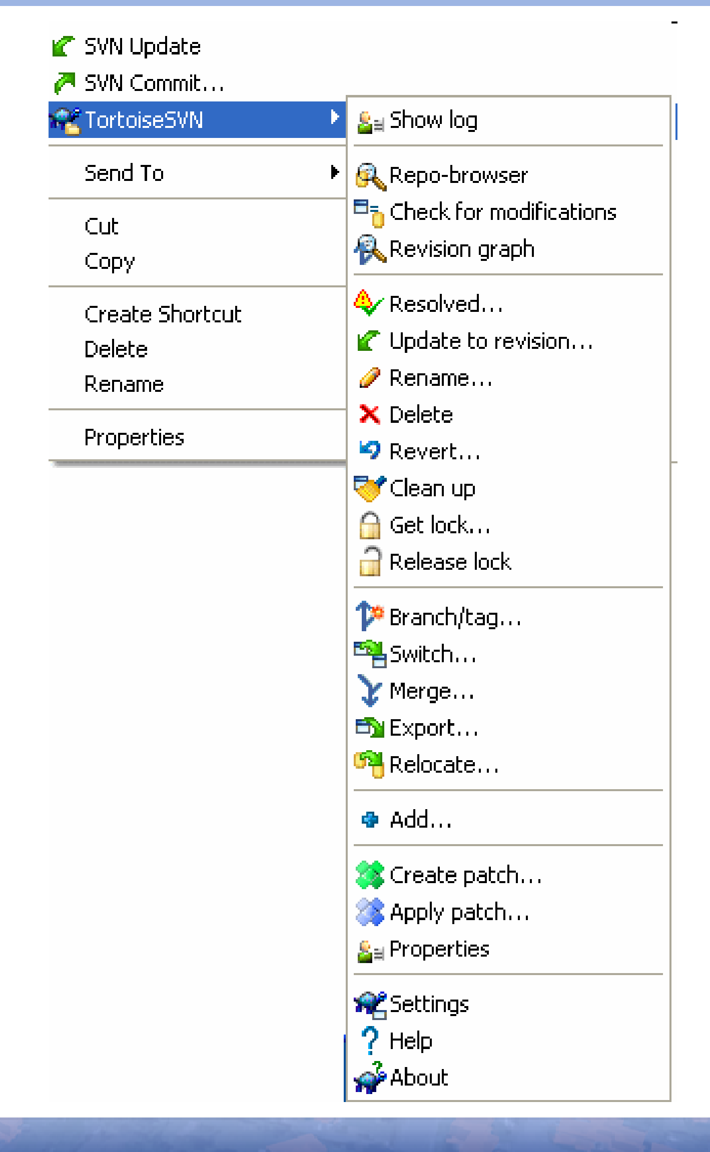


SUBVERSION



TortoiseSVN





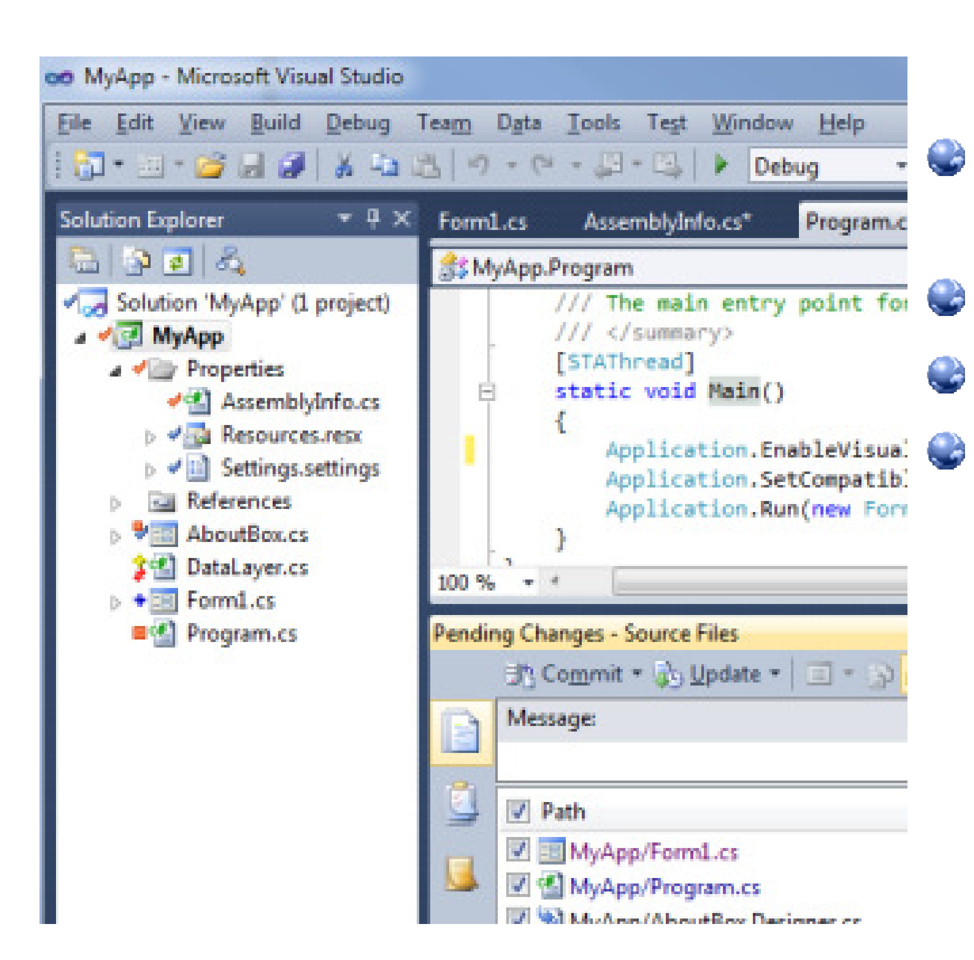
Dépôt

Écriture Lecture Lecture

Client Client Client

Source: Figure 1.1, http://svnbook.red-bean.com/nightly/fr/svn.basic.repository.html

AnkhSVN sans hésitation



- Client de contrôle pour SVN pour Microsoft Visual Studio 2005, 2008 ou 2010.
- Pour tous les types de projets de MS VS
- 100% open source
- Cité/conseillé par le projet SVN

La gestion de versions, conseils.

1. Ne pas utiliser le versionnement comme un outil de backup
2. Commiter à chaque changement d'unité logique
3. Être précis et exhaustif dans ses commentaires
4. Ne jamais casser le build du trunk (l'application doit compiler)
5. Ne créer une branche/version que si nécessaire
6. Commiter souvent et tôt

Documentation du code, quoi ?

Deux points d'entrée:

- **Documenter les attributs, propriétés, fonctions, procédures, classes, namespaces:**
  - Apporte l'intellisense (assistance au codage),
  - Génération automatique de documentation technique,
  - Utilisation des tags de documentation.
- **Documenter dans les fonctions et procédures:**
  - Explicite le code 'réellement' écrit,
  - Ni trop, ni trop peu… ,
  - Correspond plutôt au raisonnement qu'au pseudo code, i.e. le pourquoi plutôt que le comment.

Documentation du code, quand ?

- **En permanence…**

Documentation du code, comment (.net)

- Basé sur xml
- Clic-droit sur l'attribut/propriété/fonction/procédure, « insérer un commentaire »

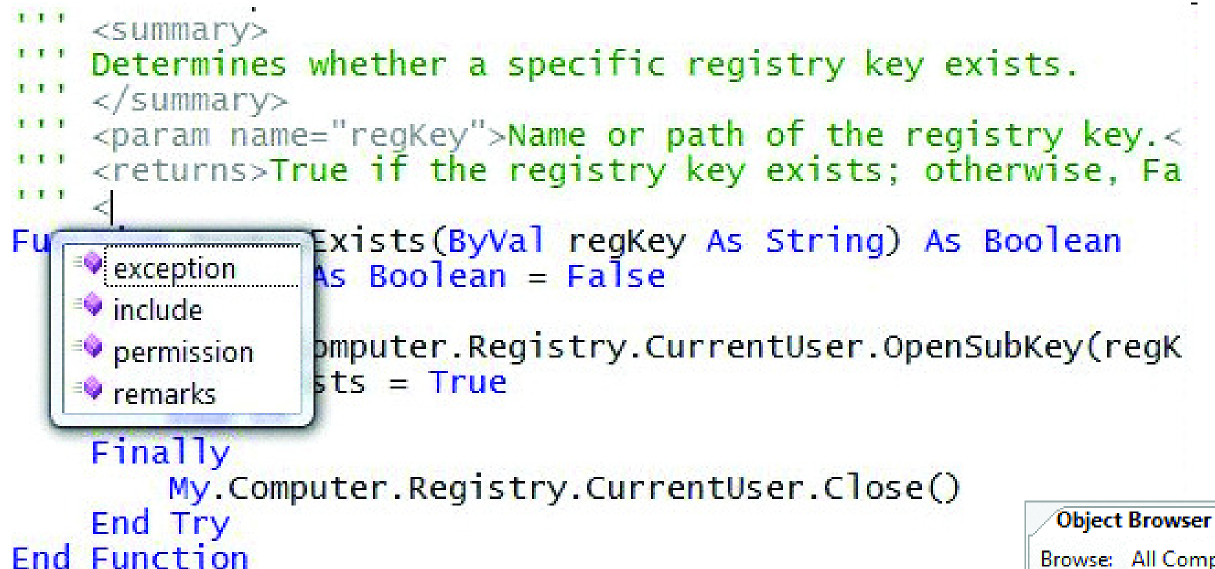
<u>OU</u>

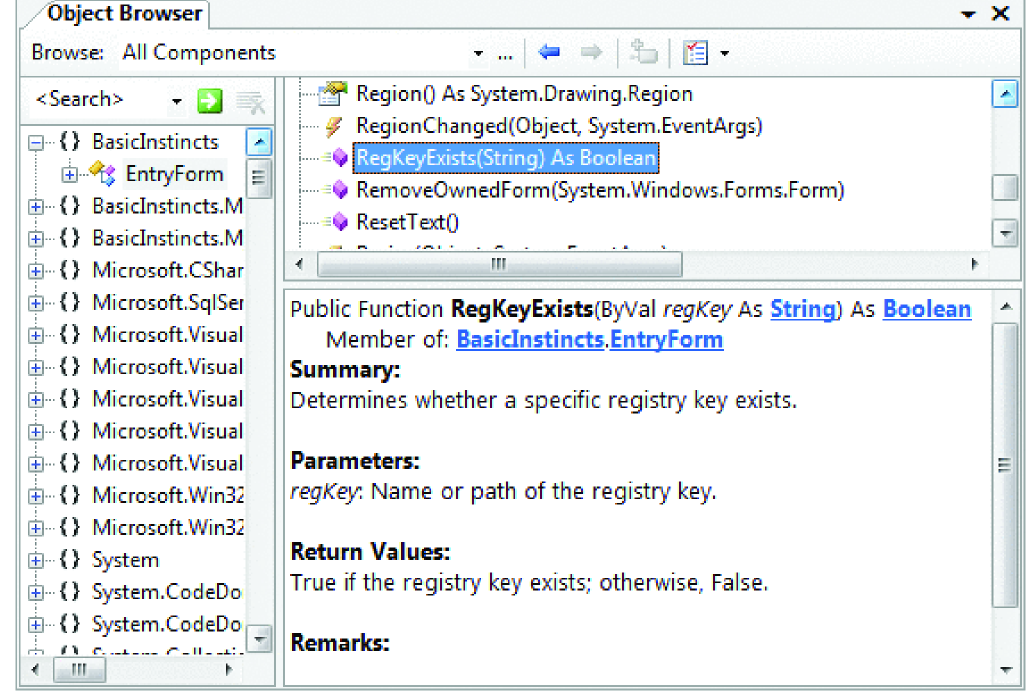
- GhostDoc
  - Clic-droit, « document this »
  - Anciennement exclusivement pour C#.Net, maintenant également pour VB.Net
  - En cas d'héritage ou d'implémentation d'interface, récupère la documentation parente
  - Interpréteur de nommage pour tenter une documentation automatique



Documentation du code, comment (JS)

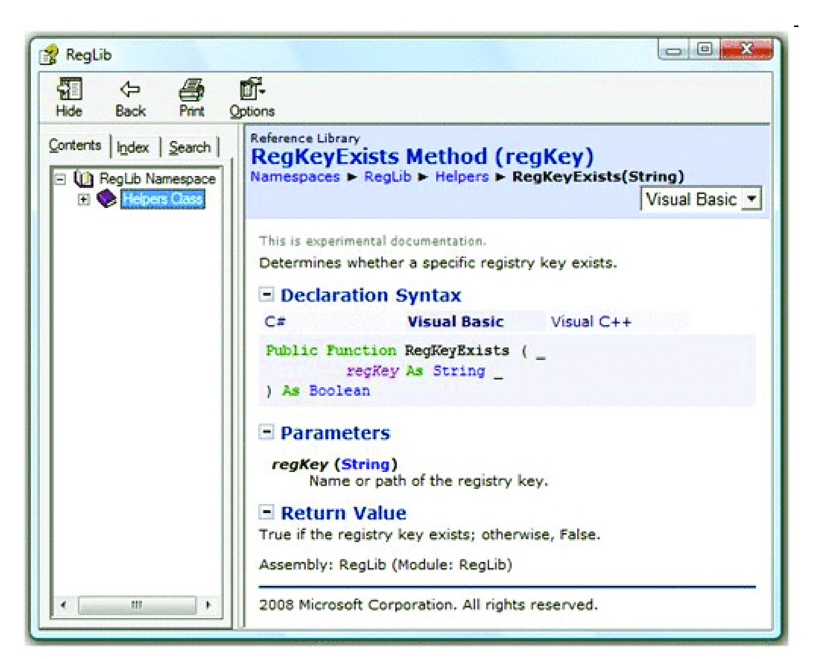
- The format for JavaScript doc comments
  - http://weblogs.asp.net/bleroy/archive/2007/04/23/the-format-for-javascript-doc-comments.aspx
- Ajouter l'intellisense sur les librairies JS:
  - http://blogs.esri.com/Dev/blogs/arcgisserver/archive/2008/09/16/How-to-get-IntelliSense-with-the-.NET-Web-ADF-JavaScript-library.aspx
  - http://weblogs.asp.net/scottgu/archive/2007/04/24/javascript-intellisense-in-visual-studio-orcas.aspx
- Using #region Directive With JavaScript Files in Visual Studio
  - http://blog.devarchive.net/2008/04/using-region-directive-with-javascript.html


''' <summary>
''' Determines whether a specific registry key exists.
''' </summary>
''' <param name="regKey">Name or path of the registry key.<
''' <returns>True if the registry key exists; otherwise, Fa
''' <|
exception
include
permission
remarks
Exists(ByVal regKey As String) As Boolean
As Boolean = False
omputer.Registry.CurrentUser.OpenSubKey(regK
sts = True
Finally
My.Computer.Registry.CurrentUser.Close()
End Try
End Function
if RegKeyExists(|
RegKeyExists (regKey As String) As Boolean
regKey:
Name or path of the registry key.


Object Browser
Browse: All Components
<Search>
BasicInstincts
EntryForm
System
Region() As System.Drawing.Region
RegionChanged(Object, System.EventArgs)
RegKeyExists(String) As Boolean
RemoveOwnedForm(System.Windows.Forms.Form)
ResetText()
Public Function RegKeyExists(ByVal regKey As String) As Boolean
Member of: BasicInstincts.EntryForm
Summary:
Determines whether a specific registry key exists.
Parameters:
regKey: Name or path of the registry key.
Return Values:
True if the registry key exists; otherwise, False.
Remarks:

Documentation générale du code, principes.

- Rassemble les classes, leur liens, héritages, leur composants…
- Souvent appelé « Library reference »
- Peut être générée automatiquement si les classes ont été commentées
  - En .chm,
  - Site web



Source: http://i.msdn.microsoft.com/dd722812.fig09_L%28en-us%29.gif

- Génération de documentation (library reference) avec Sandcastle Help File Builder
- Remplir SandcastlePath dans le projet ..\Program Files\SandCastleEcosystem\SandCastle\
- Pas de génération dans un dossier « système », y compris sur le bureau
- Peut être long
- Tout un tas de paramètres : en-tête, logo, inclusion ou non des références, ….
- Retenir qu'une fois le code documenté, l'essentiel est fait…

Optimisation du code

- Modifier du code existant
- Sans en modifier le comportement,
- Dans le but de réduire:
  - Le temps d'exécution d'une fonction,
  - L'espace occupé par les données et le programme,
  - La consommation d'énergie,
  - …

- Pas trop tôt, aux environs de la bêta

Refactoring, principes.

- Modifier du code existant
- Sans en modifier le comportement,
- Dans le but d'en faciliter la maintenance:
  - Méthode trop longue,
  - Nommage inconsistant
  - Code mort
  - …
- Généralement sous le contrôle de tests unitaires.

- Quand ?
  - Projet Agile: une fois par cycle ?
  - Autre: interval régulier…

Refactoring, comment ?

- Refactor! for Visual Basic 2008, inclus dans coderush xpress

- Notamment:
  - Changement des signatures
  - Aide sur les expressions
  - Propositions de changements

- Bonus: aide à la lecture du code
  - Renforcement de la lisibilité de la structure du code via l'emploi de lignes de tabulations
  - Soulignement des occurences au survol d'un objet

Refactoring, l'outil Refactor!.

```
private void PaintLines(Graphics graphics, int left, int top, int width, int count)
{
  int end;
  int start;
  Pen linePen = new Pen(Color.PowderBlue);
  spaceBetweenLines = 8;
  if (count >= INT_MaxCount)
    end = INT_MaxCount;
  else
    end = count;
  int y = top;
  for (int i = 0; i < end; i++)
  {
    graphics.DrawLine(linePen, left, y, left + width, y);
    y += spaceBetweenLines;
  }
}
```

```
string startingFolder = @"\Desktop\Media\Images";
```

Extract String to Resource
Introduce Constant
Introduce Constant (local)
Create new resource file
Resources.resx
Extract String to Resource
Extracts this string to a resource file.

```
string GetCorrectFileName(string file)
{
  if (string.IsNullOrEmpty(file))
    return string.Empty;

  string correctedName = file;

  while (correctedName.Contains("\\"))
    correctedName = correctedName.Remove(0, 1);

  string extension = Path.GetExtension(correctedName);
  if (string.IsNullOrEmpty(extension))
    return AssemblySearcher.FindByProjectName(correctedName);

  if (extension.Contains("proj"))
    foreach (ProjectElement proj in CodeRush.Source.ActiveSolution.AllProjects)
      if (proj.FullName.EndsWith(correctedName))
        return AssemblySearcher.FindByProjectName(proj.Name);

  return file;
}
```

```
public MediaPlayerPro(int columns, int rows, Grid parentGrid)
{
  _NumColumns = columns;
  _StartingFolder = startingFolder;
  for (int c = 0; c < columns; c++)
    parentGrid.ColumnDefinitions.Add(new ColumnDef
  for (int r = 0; r < rows; r++)
```

object name
Rename
Add Parameter
Decompose Parameter
Reorder Parameters
Add Parameter
Adds a new parameter to a method declaration and updates all calls accordingly.

Loguer, quoi ?

- Enregistrer des évènements
  - Instanciations de classes
  - Inputs/outputs fonctions
  - Valeurs de variables
  - …
- Avec des niveaux de log:
  - Debug
  - Fatal
- Dans une BD (fichier ou sgbd voire mails)
- Pourquoi ?
  - Débugger
  - Informer (bug, action réalisée, …)
  - …

Loguer, quand ?

- Dès le début du projet
- Selon ses besoins, avec un minimum syndical ☺

- **INFO Level**
  - The start and end of the method
  - The start and end of any major loops
  - The start of any major case/switch statements
- **DEBUG Level**
  - Any parameters passed into the method
  - Any row counts from result sets I retrieve
  - Any datarows that may contain suspicious data when being passed down to the method
  - Any "generated" file paths, connection strings, or other values that could get mungled up when being "pieced together" by the environment.
- **ERROR Level**
  - Handled exceptions
  - Invalid login attempts (if security is an issue)
  - Bad data that I have intercepted forreporting
- **FATAL Level**
  - Unhandled exceptions.

Loguer, comment ?

- Introduction à log4net : http://lutecefalco.developpez.com/tutoriels/dotnet/log4net/introduction/
- En contexte ESRI :http://blog.davebouwman.com/index.php/2009/10/using-log4net-in-arcmap-com-extensionscommands-etc/
- log4javascript. :http://log4javascript.org/index.html
- Fédérer les logs: TraceTool 12.1: The Swiss-Army Knife of Trace: http://www.codeproject.com/KB/trace/tracetool.aspx
- Lire les logs en direct sans verrouiller le fichier: baretail
  - http://viswaug.wordpress.com/2008/09/14/a-better-way-to-read-arcgis-server-logsand-some-weird-errors-being-logged/
  - http://www.baremetalsoft.com/baretail/

Tests unitaires, quoi ?

- Ecrire du code pour vérifier du code
- S'assurer du fonctionnement d'une partie du logiciel
- Répétable
- Vérifie l'impact d'une modification du code sur le reste du logiciel
- Argument qualité
- Aide au suivi de l'avancement

Source: http://www.esri.com/news/arcuser/0110/noyles.html

Chez ESRI:
ESRI DS2010 Developer-to-Developer:
The ESRI Development Process
http://proceedings.esri.com/library/userconf/devsummit10/papers/tech/dev-to-dev_testing_2010_devsummit.pdf

Tests unitaires, quand ?



- Si:
  - Chef de projet confirmé
  - Equipe aguerrie
  - Méthode plutôt Agile

- Alors… dès le début du projet



Tests unitaires, cas particulier: le pattern de test "objet simulacre" (Mock Object)

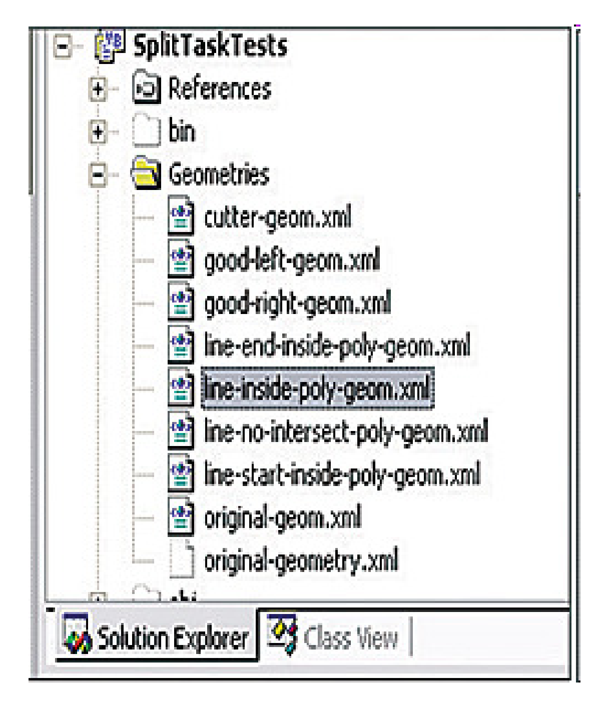
- Comment faire pour tester une méthode qui utilise une base de données ?
- Comment tester la validité d'une écriture dans un log ?
- De façon générale, comment peut-on tester du code faisant intervenir un objet pour lequel l'initialisation est plus importante que le test que l'on pensait effectuer ?

- Mock, pattern de test « objet simulacre »:
  - Un "objet simulacre" est un
  - objet qui va remplacer un des objets réels de notre solution,
  - mais qui va retourner un résultat que l'on va prédéfinir.



Tests unitaires, cas particulier: le pattern de test "objet simulacre" chez ESRI

- Seule solution pour être général et ne pas tout réinventer : ArcUnit pour pallier à la déficience ESRI dans le SDK



- Pas de changements sur le sujet à venir de la part d'ESRI

https://c.na1.visual.force.com/apex/ideaView?id=087300000086uy «Thanks for the feedback but at this time ESRI has no plans to provide a test framework. – Rob, ArcGIS Engine Product Manager »

Bien débuter son développement pour ArcGIS,

Partie 2, Professionnaliser son code

Contexte

Phase de développement: des principes

Phase de développement: des pratiques



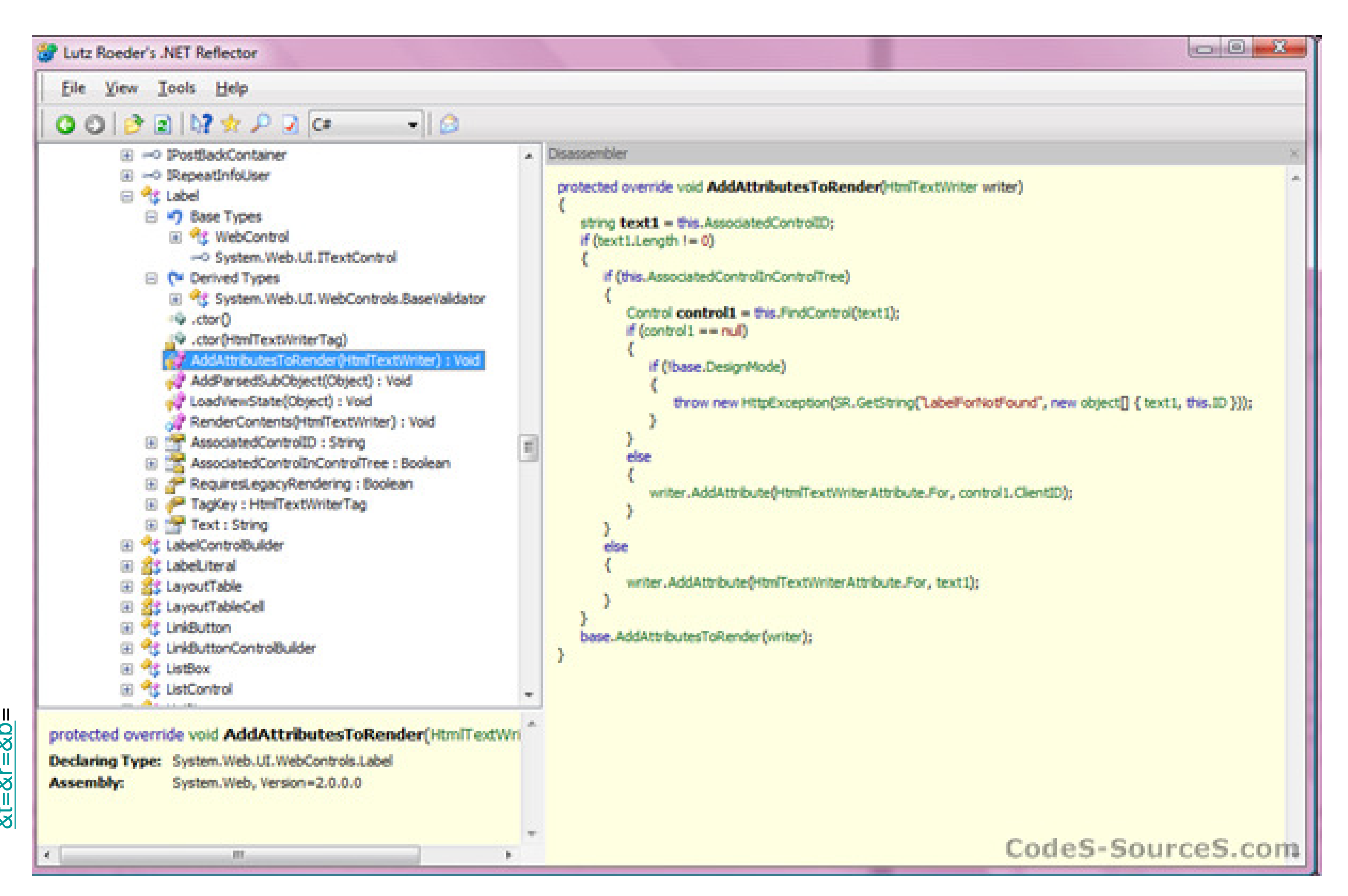
Phase de développement: des outils généraux (et gratuits)

Conclusion

Reflector, principes.

1. Dé-compile des exe ou des dll du framework .Net
2. Permet de pallier à une absence de doc
3. Peut permettre de comprendre le code utilisé par une librairie tierce

Reflector, en image.



Source: http://www.csharpfr.com/tutoriaux/imageHandler.ashx?documentID=108&name=/word/media/image8.png&w=5761990&h=3756660&l=&t=&r=&b=

- Parmi la liste des add-ins pour reflector disponibles, citons notamment:

- Refexil, qui permet la modification du code d'une assembly (dll),
- FileDisassembler, qui permet d'exporter/désassembler dans le langage de son choix une assembly,
- AutoDiagrammer et SequenceViz, qui schématisent les classes et séquences de l'assembly.

- Liste complète: http://reflectoraddins.codeplex.com/

FxCop et StyleCop, principes.

- Conformité aux "bonnes pratiques" recommandées par Microsoft.

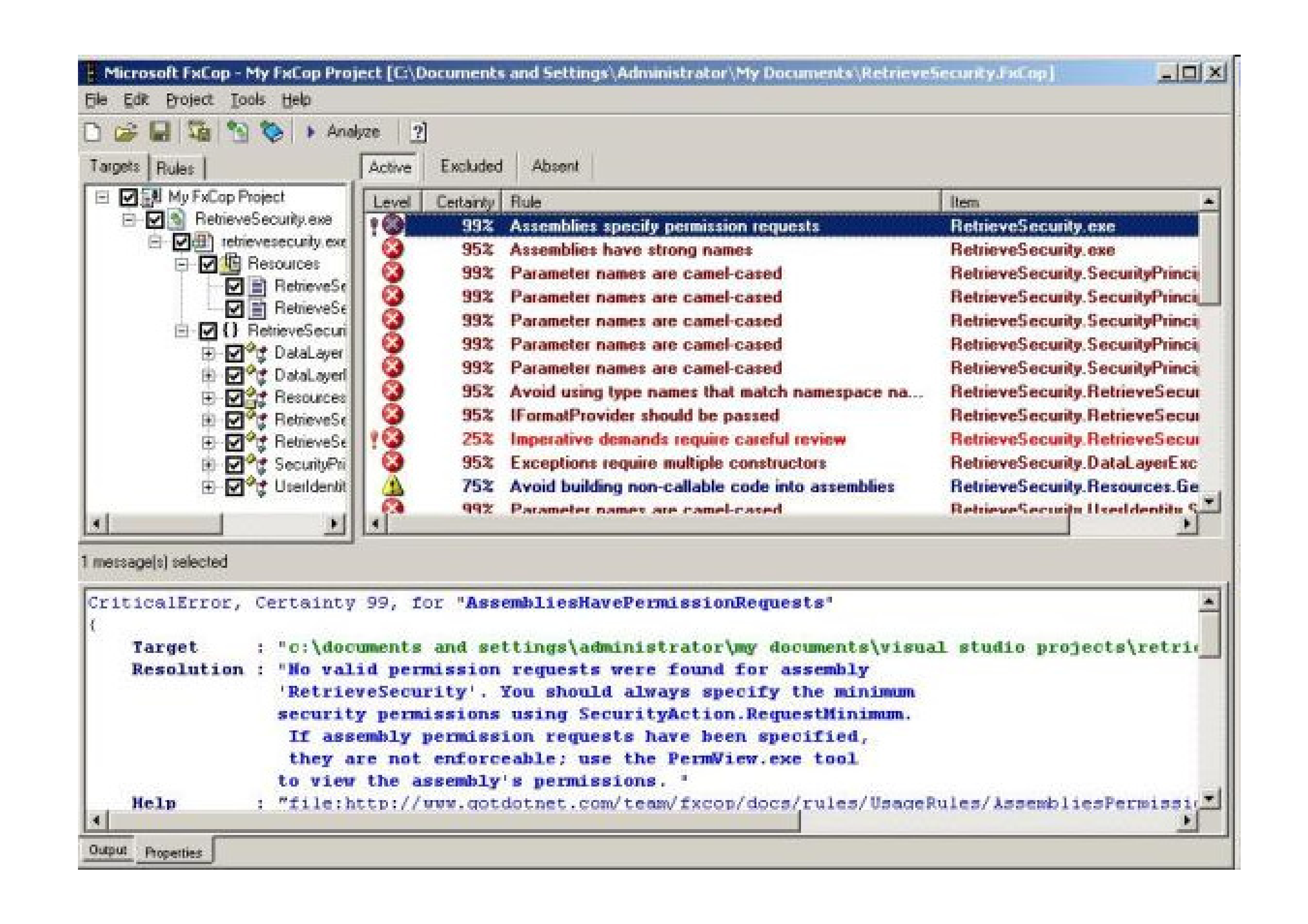
- Design Guidelines for Class Library Developers

http://msdn.microsoft.com/en-us/library/czefa0ke%28VS.71%29.aspx

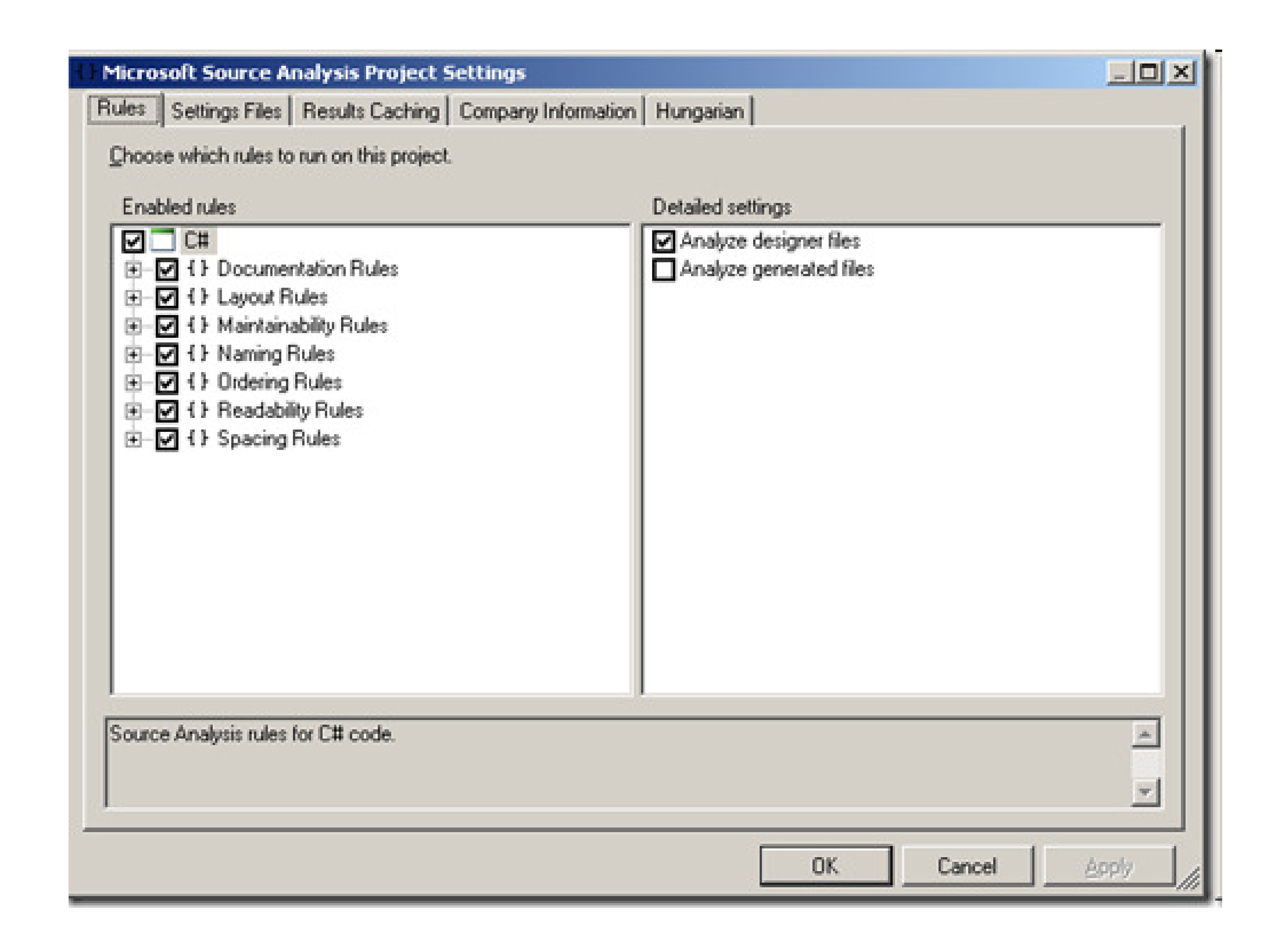
- FxCop
  - Plutôt binaire
  - Plutôt sécurité, bonne pratiques de code
- StyleCop
  - Plutôt code source
  - Plutôt bonnes pratiques de présentation
- Attributs insérables (sur namespace, classe, fonction, …) pour indiquer que l'on déroge à une règle en toute conscience.

FxCop et StyleCop, FxCop

- Design de la librairie
- Globalisation (langues)
- Conventions de nommage
- Performance
- Interopérabilité
- Portabilité
- Sécurité
- Usage



- Documentation
- Modèle, présentation
- Maintenance
- Nommage
- Ordonnancement
- Facilité de lecture
- Espacement





NArrange - .NET Code Organizer/Formatter/Beautifier

- Ré-arrange le code
- En accord avec ses conventions et styles
- (#region, attributs et propriétés groupées (ou non!), …

DemoClass.cs — DemoProject.DemoNamespace.DemoClass — DemoClass()

```
// Demo class for NArrange
//
// Copyright/revision information here.
using System;
using DemoProject;
using System.IO;
using System.Text;
namespace DemoProject.DemoNamespace
{
    /// <summary>
    /// NArrange demo class
    /// </summary>
    public class DemoClass
    {
        // Default constructor
        public DemoClass()
        {
            // Do something here
        }
    }
}
```

DemoClass.cs — DemoProject.DemoNamespace.DemoClass — DemoClass()

```
using System;
using System.IO;
using System.Text;

using DemoProject;

namespace DemoProject.DemoNamespace
{
    /// <summary>
    /// NArrange demo class
    /// </summary>
    public class DemoClass
    {
        #region Constructors

        // Default constructor
        public DemoClass()
        {
            // Do something here
        }

        #endregion Constructors
    }
```

Region Added



Compteur de lignes de code

C#, Visual Basic and C++ .NET Line Count Utility

Line Count

File Options Help

C:\csharp\Med.sln:
Number of lines = 180,201, number of code-generated lines = 99,943, number of user-entered blank lines = 9,586, number of user-entered comments = 9,478

Project Name	Project Path	Project - Lines	Project - Blank Lines	Project - Designer Lines	Project - Comments
Presentation.csproj	Presentation\Presentation.csproj	54,894	3,745	21,990	3,671
DatabaseManager.csproj	DatabaseManager\DatabaseMana...	2,302	280	0	424
DataAccess.csproj	DataAccess\DataAccess.csproj	77,219	435	73,493	361
Common.csproj	Common\Common.csproj	1,137	69	0	213
Model.csproj	Model\Model.csproj	21,440	2,352	1,528	1,930
UnitTest.csproj	UnitTest\UnitTest.csproj	15,877	2,150	0	2,235
TestInstall.csproj	TestInstall\TestInstall.csproj	252	16	57	68
AnalyticsWebService.csproj	AnalyticsWebService\AnalyticsWe...	463	46	0	68
ErrorModel.csproj	ErrorModel\ErrorModel.csproj	3,320	399	260	338
AnalyticsWrapper.csproj	AnalyticsWrapper\AnalyticsWrapp...	509	82	0	74
AuthenticationService.csproj	AuthenticationService\Authenticati...	2,697	6	2,615	41
StartUp.csproj	StartUp\StartUp.csproj	91	6	0	55

Conversion C# <-> VB

- SharpDevelop (IDE .Net Open Source)
  - http://www.icsharpcode.net/opensource/sd/
  - + = Conversion complète de la solution possible!
- Service web (basé sur SharpDevelop):
  - http://www.developerfusion.com/tools/convert/csharp-to-vb/
  - http://www.developerfusion.com/tools/convert/vb-to-csharp/
- Plug-in VS CodeConvert (basé sur SharpDevelop)
  - http://codeconvert.codeplex.com/



Revue de code: Code review plug-in for Visual Studio 2008

- Faire des commentaires sur du code
- Enregistrés dans un xml associé à un fichier de code
- Et donc diffusable pour revue/discussion collaborative



- En commentaire, inscrire TODO

```
29 + Read Methods
60
61 - #Region "Add/Update/Delete Methods"
62       ' TODO: Fill these in, when needed or before the end of the month
63 - #End Region
64
65 - #Region "Hydration Methods"
```

Task List - 1 task

Comments

!	Description	File	Line
	TODO: Fill these in, when needed or before the end of the month	DataController.vb	62

Error List | Task List | Output

Source: http://kb.gilleland.info/Home/tabid/808/ctl/ArticleView/mid/1674/articleId/159/PageID/176/Comments-and-Tasks-in-Visual-Studio.aspx#axzz0usptP8jN

- Autres tags ------------------->

Options

- Environnement
  - Général
  - Aide
  - Clavier
  - Démarrage
  - Documents
  - Importation et exportation de
  - Liste des tâches
  - Navigateur Web
  - Paramètres internationaux
  - Polices et couleurs
  - Rechercher et remplacer
  - Récupération automatique
  - Sécurité des compléments/mac
- Projets et solutions
- Contrôle de code source
- Éditeur de texte

Options de la liste des tâches
- [x] Confirmer la suppression des tâches
- [x] Masquer les chemins de fichiers complets

Jetons

Liste des jetons :
- HACK
- TODO
- UNDONE
- ! UnresolvedMergeConflict
- UPGRADE_TODO
- UPGRADE_WARNING

Priorité : Normale

Nom :

Ajouter | Supprimer | Modifier

- [x] Afficher tous les paramètres

OK | Annuler

Bien débuter son développement pour ArcGIS,

Partie 2, Professionnaliser son code

Contexte

Phase de développement: des principes

Phase de développement: des pratiques

Phase de développement: des outils généraux (et gratuits)



Conclusion

Conclusion

- Au travers de cette présentation, le géomaticien débutant en développement pour Arcgis possède les clés permettant de de dialoguer ou de collaborer sereinement avec un développeur professionnel…

- Et qui sait, peut être même se sentir capable d'internaliser le développement ?





Merci

Le Doc
Lud
Warg
David
ClaireLK

Christelle
Benoît
Et les autres

Pour leurs relectures et commentaires



